Sensor use and usefulness: Trade-offs for data-driven authentication on mobile devices

Sensor Use and Usefulness: Trade-Offs for Data-Driven Authentication on Mobile Devices

(In proceedings of PerCom 2015, ©IEEE)

Nicolas Micallef*, Hilmi Güneş Kayacık†, Mike Just‡, Lynne Baillie* and David Aspinall§
*Glasgow Caledonian University, Glasgow, Scotland, {nicholas.micallef, lynne.baillie}@gcu.ac.uk
†FICO, 181 Metro Dr., San Jose, CA, guneskayacik@fico.com
‡Heriot-Watt University, Edinburgh, Scotland, m.just@hw.ac.uk
§University of Edinburgh, Edinburgh, Scotland, david.aspinall@ed.ac.uk

***Abstract*—Modern mobile devices come with an array of sensors that support many interesting applications. However, sensors have different sampling costs (e.g., battery drain) and benefits (e.g., accuracy) under different circumstances. In this work we investigate the trade-off between the cost of using a sensor and the benefit gained from its use, with application to data-driven authentication on mobile devices. Current authentication practice, where user behaviour is first learned from the sensor data and then used to detect anomalies, typically assumes a fixed sampling rate and does not consider the battery consumption and usefulness of sensors. In this work we study how battery consumption and sensor effectiveness (e.g., for detecting attacks) vary when using different sensors and different sensor sampling rates. We use data from both controlled lab studies, as well as field trials, for our experiments. We also propose an adaptive sampling technique that adjusts the sampling rate based on an expected device vigilance level. Our results show that it is possible to reduce the battery consumption tenfold without significantly impacting the detection of attacks.**

I. Introduction

Mobile devices have increasingly become a part of our daily lives. Their usefulness not only stems from their computing power but also their collection of sensors which allow rich interaction between the device, the user and the environment. However, an ever-present trade-off exists between the cost of using a sensor and the benefit gained from its use. If all sensors were to be switched off, then the battery life of a mobile device would last for weeks without need for charging, but this would lead to a considerable reduction in features. Certainly, the battery capacity of mobile devices has been increasing steadily. For instance, the standard battery for Samsung Galaxy S series phones are 2100 mAh, 2600 mAh and 2800 mAh for S3, S4 and S5 respectively.[1] However, this increase is often matched by increased usage and resource requirements [1]; some users charge their devices more than once a day [2] although this is hardly convenient. Therefore, for numerous applications it is important to understand the costs and benefits of mobile device sensors.

Sensor data is often used in mobile applications to provide location and context awareness, triggering specific actions or offering the user relevant choices. Sensor data can also be used to fingerprint user behaviour, enabling continuous *data-driven authentication*, to ensure that the device is being used by its proper owner: if the device detects an anomalous behaviour, it can take action to explicitly authenticate the user or raise an alarm. Previous work (e.g., [3], [4], [5], [6], [7]) has shown that data from a broad range of sensors can be used to provide quite accurate authentication, but researchers have generally not considered the cost of doing this, nor compared the relative effectiveness of the various signals available. Work so far on sensor power consumption (e.g., [8], [9], [10]) has focused mainly on minimising use of 'high drain' sensors (such as GPS), though only in terms of resource consumption and not security protection. For practical security applications, we must find an acceptable balance between power consumption and dependable security alerts. For example, reliable authentication depends on frequent sampling, which is costly: what effect does a decrease in sensor sampling frequency have on resource consumption and security? Our goal is not to perform a rigorous analysis of different mobile device hardware capabilities (batteries and sensors), but rather to establish a benchmark regarding the cost and benefit trade-offs for using sensors for data-driven authentication on some of today's devices.

Our contribution is two-fold:

1) We examine the impact of mobile device sensors in terms of battery consumption and authentication effectiveness, under variations of sensor sampling frequency and individual sensor contributions.
2) Drawing from our analysis, we propose an adaptive sampling technique that adjusts the sensor sampling rate according to several factors, based upon an expected device vigilance level.

For our studies we make use of data from controlled experiments using two Android devices, as well as data from field studies of real mobile device usage.

In Section II we review the related research on sensor optimisation as well as on data-driven authentication. In Section III we empirically compute the battery consumption per sensor for different rates of sampling. In Section IV, we analyse the effectiveness of *all* sensors for detecting attacks for different sampling rates, as well as the contribution of *each* sensor for attack effectiveness. An adaptive sampling technique is proposed in Section V with results on its efficacy and resource consumption. Conclusions and future work are discussed in Section VI.

[1] Retrieved from www.samsung.com, 20 Jan 2015, for the standard batteries EB-L1G6LL, EB-B600BUB and EB-BG900BBU.

II. Related work

Previous work on sensor resource consumption investigated the battery cost of mobile device sensors mainly to minimise the use of 'high drain' sensors such as GPS, and also looked into innovative techniques such as shared caching, speculative sensing and adaptive sampling. However, optimising sensors for security has not been considered. Work on data-driven authentication has used a variety of sensors for modelling user behaviour though has not considered individual sensor impact on security effectiveness or resource consumption.

A. Research on resource consumption

Optimising resource consumption during mobile sensing has been a popular topic of discussion in the area of context-aware computing, mostly due to the increasing availability of rich sensors on today's mobile devices [8], [9]. Much of the context-aware research focuses on improving resource consumption by exploiting redundancy, using low power sensors. Paek et al. [11] and Zhuang et al. [12] use the accelerometer to define which localisation techniques to use, and to adjust sampling rates based on the battery level. Lin et al. [8] and Wang et al. [13] use low power sensors to detect user states and context, and trigger high power sensors only when required. With MobiSens, Wu et al. [10] reduce the GPS sampling rate when a user is not moving. Schirmer at al. [14] took a step further, by generalising the technique of improving resource consumption by substituting a high drain sensor with a semantically related but low drain sensor.

Sensor optimisation techniques by Li et al. [15] used machine learning algorithms to improve the energy efficiency of multiple high drain sensors by trading off the sensing accuracy. They reported that their technique improves resource consumption while at the same time keeping the sensing accuracy higher than 90%. Other approaches exploit redundancy across applications by sharing sensor data and inferred context attributes among applications through a shared cache [16], which also has the limitation of still being high drain if applications involve expensive sensors such as GPS and microphone. ACE [17] proposed an improvement, which dynamically learns relationships among various context attributes through the use of speculative sensing (inferring the value of a high drain attribute by sensing lower drain attributes). Nath et al. [17] claim that ACE can reduce battery consumption by about 4.2 times, compared to a raw sensor data cache shared across applications.

Selective sampling is also a popular approach used to optimise resource consumption. Krause et al. [18] show that by using optimised selective sampling schemes they could increase the deployment lifetime of their eWatch wearable platform by a factor of four without a substantial loss in prediction accuracy. Following on this research Rachuri et al. [19] came up with an adaptive sensor sampling methodology which relies on dynamic selection of sampling functions depending on history of context events. They show that a dynamic adaptation mechanism provides better trade-offs compared to simpler function-based rate control methods.

In our research, we also consider sensor optimisation, though we are motivated to optimise security effectiveness as well as battery consumption. Our use of adaptive sampling is related to that described above.

B. Research on data-driven authentication

Profiles built from sensor data can subsequently be used to identify a user for authentication or access control purposes, if the sampled sensor readings are consistent with the profile.

Gupta et al. [3] proposed a model for the familiarity and safety of a user's device based upon its location, and used this to automatically construct access control policies. Their model distinguishes the behaviour of different users, and incorporates user feedback for refinement, though they do not investigate the contribution of the sensors used in their model. Shi et al. [4] focused on implicit authentication by learning user behaviour and assigning a score – positive for familiar events and negative for unusual – based on recent user activity. They show the power of fusing multiple features together though they do not investigate which are the features that contribute the most to their algorithms. Lin et al. [20] proposed a non-intrusive authentication method based on orientation sensor data using k-nearest neighbour classification. Despite having the limitation of focusing on just one sensor, Lin et al. [20] argue that while input from a single sensor may yield poor accuracy, combining multiple sensor inputs would improve the accuracy. To this end, Senguard [21] aimed to implicitly and continuously authenticate users using input from many sensors yielding a stronger classifier built from per-sensor classifiers. This research introduces the concept of using different sensor modalities based on which context the user is currently in, though they do not evaluate the contribution of the sensors to the accuracy achieved by each modality.

Kayacik et al. [7] describe a spatial and temporal model for building user profiles from sensor data that is data-driven and automatically builds profiles, sets thresholds and detects behaviour drift. Despite using a variety of sensors (such as wifi networks, cell towers, application use, etc.) from three different datasets, they do not investigate the contribution of each sensor. Furthermore, context aware authentication research [6], [22], [23], [24] focused on sensing the context in which the device is used (such as home or work) and providing access based on device comfort computed from various sensor data.

Our research explores the effectiveness of the above sensor-based models in terms of each sensor's contribution to an accurate authentication and to resource consumption.

III. Battery Consumption

Previous research on battery consumption distinguished between high drain and low drain sensors [13], [14] but very few [8] focused on quantifying the battery consumption of sensors on mobile devices. Battery consumption depends on a number of factors, including implementation decisions, hardware specifications and the signal quality (e.g., Wi-Fi and GPS). Here, we aim to empirically study the battery usage of sensors using a benchmark data collection app. Hence, our goal is to demonstrate relative differences in individual sensor resource consumption at different sampling rates. To limit the variability of our study, we focus in this section on sensor cost only, and introduce the security detection tool in Section IV. Trade-offs between security and consumption are also considered.

A. Method

For our battery consumption experiments, we performed a controlled lab study using 2 dedicated Samsung Galaxy S4 phones with 2GB RAM, 1.9 GHz Snapdragon 600 processors, Li-Ion 2600 mAh battery, running Android 4.2.2. The devices were configured with identical settings. Automatic updates, location services and other features that may generate activity are turned off. To ensure the devices are sensing data from real life conditions (especially for GPS which experiences a significant change in behaviour when used indoors rather than outdoors) but at the same time experiencing the same external conditions, the experimenter carried the devices together throughout the duration of the studies. Using this setup the devices experienced a full 4 days of the experimenter's 'daily routine.'

To collect our experiment data, we used a tool with two components (we add a detector component in Section IV): a sensor data collector (*collector*), and *PowerTutor* [25] for measuring the battery consumption. Zhang et al. [25] claim that the power consumption estimates given by *PowerTutor* app on HTC G1, HTC G2 and Nexus phones are within 5% of actual values and that the average long-term error is less than 2.5% over the application's life span. The *collector* was developed using Android SDK 4.2.2, and was tested on a set of our research group's lab phones. Before conducting these experiments we used the *collector* on a wide range of Android devices to collect data for some of our related studies.

The *collector* used in this experiment is configured with the following settings. The microphone, accelerometer and rotation sensors are configured to collect data for 5 seconds. The light and magnetic field sensors are configured to collect 15 samples per reading (meaning that if the sampling rate is set to 1 minute, with this configuration the system will collect 15 samples every minute). The app usage is configured to execute a linux *top* command (which gets a detailed snap shot of all processes currently running on the device) and saves the returned values to a text file. The Wi-Fi sensor retrieves a list of the detected Wi-Fi networks and cell towers each time that it is triggered. Similarly, the GPS sensor is configured to retrieve the current location and save them to disk each time that this sensor is triggered.

For every second, *PowerTutor* collects the total mW consumed by the *collector*. The total mW is divided by the voltage to obtain the mA consumed for each second. Subsequently, we compute the mA consumed by the app during every hour of this study (mAh), which we then add together and divide by 96 hours to have the average mAh consumed by the *collector* throughout the 4 days of the study. We apply this methodology for each of the mAh values reported in Tables I and III.

B. Consumption for different sampling rates

To understand how resource consumption changes with different sampling rates, we configured the *collector* to collect accelerometer, magnetic field, light, rotation, Wi-Fi, app usage, microphone and GPS data. The battery consumption values reported in Table I represent the average battery consumed in an hour by the *collector* during the 96 hours in which the particular sampling rate was being used.

TABLE I. AVERAGE BATTERY CONSUMPTION PER HOUR (MAH) CONSUMED BY THE COLLECTOR UNDER DIFFERENT SAMPLING RATES.

Rate	Battery (mAh)
1 min	10.83
5 min	2.72
10 min	1.04
15 min	0.71
20 min	0.45

As expected, the battery consumption results reported in Table I show that as the sampling rate decreases there is a proportional drop in battery consumption. Falaki et al. [26] report that light drain use consumes about 10 mAh, medium drain use consumes about 90 mAh and high drain use consumes about 250 mAh. We use these drain values together with the battery consumption results reported in Table I to compute an estimated time taken to drain the device (in hours) when running the *collector* using different sampling rates (refer to Table II). For these estimates we do not consider the night period when the phone is often idle. To calculate the time taken to drain the device when running the *collector* using a 1 minute sampling rate with light drain use, for example (as reported in row 3, column 2 of Table II), we assume a battery capacity of 2600 mAh [1], 10 mAh of light drain use and 10.83 mAh for 1 minute sampling. We divide the battery capacity with the result of the addition of the light drain use to the battery consumption for the 1 minute sampling rate:

$$\frac{2600 \text{ mAh}}{10 \text{ mAh} + 10.83 \text{ mAh}} = 124.80 \text{ hours} \quad (1)$$

TABLE II. PRACTICAL REAL-LIFE EXAMPLES OF TIME TAKEN TO DRAIN THE DEVICE WHEN RUNNING THE COLLECTOR USING DIFFERENT SAMPLING RATES. () = REDUCTION IN BATTERY LIFETIME COMPARED TO BASELINE.

Rate	Light	Medium	High
Baseline	260.00h	28.89h	10.40h
1 min	124.80h (52.0%)	25.79h (10.7%)	9.97h (4.1%)
5 min	204.39h (21.4%)	28.04h (2.9%)	10.29h (1.1%)
10 min	235.30h (9.5%)	28.55h (1.2%)	10.36h (0.4%)
15 min	242.79h (6.6%)	28.66h (0.8%)	10.37h (0.3%)
20 min	248.85h (4.3%)	28.74h (0.5%)	10.38h (0.2%)

The baseline (row 2) in Table II reports the time taken to drain the device when the *collector* is not installed on the device. The percentages next to each value show the percentage reduction in battery lifetime of the particular sampling rate when compared to the baseline. From these results, we can see the impact of the different sampling rates on battery lifetime, especially for the low drain and medium drain uses. For example, if we assumed that users might tolerate a 10% reduction in their battery lifetime to support data-driven authentication, then 1 minute sampling for both the low drain and medium drain uses would be too costly, and even 5 min sampling for the low drain use would be too costly. These results also indicate that high drain uses would not experience a significant reduction in battery lifetime when low sampling rates are used.

C. Consumption of different sensors

In order to measure the battery usage per sensor, we fixed the sampling rate at 1 minute and ran the *collector*, activating one sensor at a time. The battery consumption values reported

in Table III represent the average battery consumed in an hour (in mAh) by the *collector* during the 96 hours in which one sensor was switched on.[2] The results reported in Table III confirm that the GPS and accelerometer sensors are the highest draining sensors.

TABLE III. AVERAGE BATTERY CONSUMPTION PER HOUR (MAH) WHEN SENSORS ARE ACTIVATED ONE AT A TIME.

Active Sensor	Battery (mAh)
Accelerometer	2.08
Apps Usage	1.46
GPS	2.31
Light	0.86
Magnetic Field	0.49
Microphone	1.71
Rotation	2.01
Wi-Fi + Cell	1.62

IV. ATTACK DETECTION

In our experiments, we use a publicly available tool for building user profiles and attack detection as a benchmark for our analysis, from which we provide some general results that would be applicable to similar data-driven authentication techniques. We study the effectiveness of attack detection, first when all sensors are activated in *userprofiler* (Section IV-C), and then when one sensor is activated at a time (Section IV-D).

A. Method

We use the publicly available *userprofiler* project [7], [27]. It is suitable for our experiments because (1) it is publicly available, (2) it allows us to activate one sensor at a time and (3) allows us to adjust the sampling rate on the fly. In this work, we use *userprofiler* as a benchmark, with minimum changes (to fit within our experimental framework) and mainly focus on sampling rates and sensor contribution. Thus, extending existing systems is not within the scope of this work.

Userprofiler builds temporal and spatial models from the observed sensor data. The resulting profile consists of a set of probability density functions built for different hour-of-day and locations (as approximated from cell towers) describing the general characteristics for that location or time through the functions as shown in Figure 1. Using the temporal and spatial models, the device assigns a score for each sensor event based on its frequency of occurrence for that location or time. Higher scores indicate high familiarity with the observed data while low scores indicates an anomaly. If the observed behaviour deviates from the norm as established by the probability density functions, the detection score drops. Subsequently, an alarm is raised and the device can potentially be locked down.

In order to evaluate the security of *userprofiler*, we employed two datasets, one containing the normal use of 4 users over a 3 week field study and the other containing attacks for one user. The attacks were generated under controlled conditions, in which the device was not actually stolen, but a participant is tasked with simulating 'misuse.' The sensor data was collected from Android devices and in addition to cell tower data for the *userprofiler* location, it included data from wifi networks, application use, light and noise levels, accelerometer, rotation, and magnetic field.

[2]Due to a constraint of our collector, the results for the radio sensors (Wi-Fi and cellular) are reported as a single value.

B. Attack model

We created two attacks with varying levels of sophistication. An *uninformed adversary* has little knowledge of the user's behaviour and has the intention of stealing the device. On the other hand, the *informed adversary* has reasonable knowledge of the user and is aware of the detector running on the device, and thus aims to keep the device in familiar surroundings and use the device to extract sensitive information. Additionally, we use one day of normal behaviour data in which a typical weekday usage behaviour (i.e., 9 am - 5 pm with commutes and without any anomalies) is captured.

To facilitate our experiments, we asked one of our participants to use their device for 3 weeks. After the training is complete and detector is deployed, we asked another participant to use the device for a few hours per scenario to create the following attack scenarios:

- **Uninformed adversary:** The attacker took the device and carried it with them for one day. The attacker lived in another city, ensuring that the locations they frequent were different from the owner's. The attacker had no additional information on how the owner uses their device. The attack started at 2 pm. We claim that this is similar to a typical device theft.
- **Informed adversary:** The attacker, who was the owner's housemate, used the device at the owner's home for the day. Thus, the location was well known to the device. The participant attacker was provided with a list of applications that the owner frequently uses. The attacker used the device between 1 pm and 5 pm. This scenario corresponds to an insider attack in which a capable insider attempts to use the device at a well-known location.

For a comparison of the attack data to **normal use**, one day of the owner's usage is utilised to measure false positive rates under normal conditions. The one-day normal data is extracted from the user's normal usage data, before creating the training data for *userprofiler*. The normal day is selected to: (1) be a weekday where the user is at work between 9 am and 5 pm; (2) contain commute to work and; (3) be validated to be anomaly free (i.e. no unusual travel and work hours).

C. Attack detection with all sensors

In order to measure detection rates, the training and the attack data is collected with a 1 minute sampling rate. To compute detection rates for lower sampling rates (e.g., 5 minutes), the data is down-sampled by sampling every N minutes (from the original 1 minute sampled data), where N is the current sampling rate. Detection results are provided in Table IV in terms of detection time and detection rate. Detection time is the time it takes, in seconds, for the detector (in the *userprofiler*) to flag the behaviour suspicious from the start of the attack. In the case of *userprofiler*, a detection occurs after four anomalous events. Detection rate on the other hand provides a percentage of samples marked as suspicious during the duration of the attack, even after the detection occurs. False positive rate provides a percentage of samples marked as suspicious during the normal use day. Computing both detection time and rate allows us to go beyond the immediate

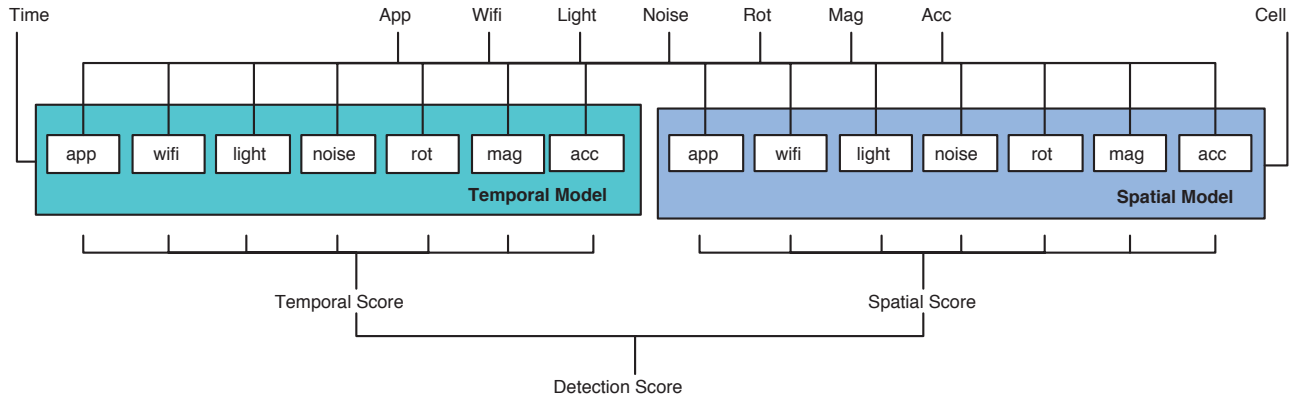


Fig. 1. Detection diagram for userprofiler.

device locking scenario and investigate what would happen if the anomalous events were logged to make a security decision based on historical data.

TABLE IV. DETECTION TIME (DT) AND RATE (DR) OF ATTACKS, FALSE POSITIVE RATE (FPR) FOR A DAY WITHOUT ATTACKS UNDER DIFFERENT FIXED SAMPLING RATES.

Rate	Uninformed		Informed		Normal
	DT	*DR*	*DT*	*DR*	*FPR*
1 min	183 s	92.07%	1657 s	28.82%	1.39%
5 min	3591 s	92.10%	6012 s	20.00%	0.72%
10 min	4790 s	92.98%	Undetected		1.45%
15 min	5406 s	96.42%	Undetected		3.26%
20 min	5987 s	95.65%	Undetected		1.47%

The results show that reducing the sampling rate does not substantially change the attack detection rates. In other words, the attacks are (almost) equally detectable under different sampling rates although lower sampling rate shows a slight increase in detection rate. This happens because lower sampling favours frequently occurring samples, i.e., the attack, in the case of the uninformed attack. However, detection time increases as the sampling rate is reduced (the non-linear increase in detection time, as a function of sampling rate, in Table IV is due to the varied distribution of the four anomalous events required for a detection). Given that detection requires an anomaly to be observed, it takes longer to detect the attacks under lower sampling rates. Compared to the battery consumption results in Table I, battery improvement comes with the expense of extending detection time over 10 times (i.e., when rate decreases from 1 min to 5 min), which indicates that fixed sampling rates over a few minutes would not be effective against attacks. The particular *userprofiler* implementation, by default, expects four events to be anomalous for detection, therefore if an attack is "undetected", it does not mean that the score never drops below the detection threshold. Rather, it means the score does not remain below the threshold long enough to be detected.

There are at least a couple of options for improving efficiency while maintaining a sufficient level of security. In Section IV-D we investigate the option of using fewer sensors. In Section V, we vary the sampling rate in different circumstances, and measure the effect on security and efficiency.

D. Attack detection with each sensor

To determine the individual sensor contributions (their 'usefulness'), we compute the detection time, detection rate and false positive rate that are observed when one sensor is activated at a time. To this end, we extend the *userprofiler* tool to selectively activate the sensors. As a result, we ran additional experiments on two attack scenarios and a normal use dataset while activating one sensor at a time. For our analysis, we defined two ad-hoc sensor combinations (others are possible). *Ambient sensors* consist of noise, magnetometer, light and wifi and capture what is happening around the device. *Behavioural sensors* consist of app usage, rotation, accelerometer and thus, capture user's use of the device. While other combinations are possible, measuring efficacy per sensor and per ambient and behavioural subsets allow us to determine not only individual sensor usefulness but also whether ambient or behavioural sensors are more useful in terms of detection rates and false positives.

Table V provides the experimental results for attack scenarios and normal use. Results are expressed in terms of detection rates, times and false positive rates, as defined in Section IV-C. An attack is undetected if the score fails to remain below the detection threshold for four consecutive events, which was the default detection rule for *userprofiler*. The last row shows the results when all sensors are activated and *amb* and *beh* shows the results for ambient and behavioural sensors, respectively.

TABLE V. DETECTION TIME (DT) AND RATE (DR) OF ATTACKS, FALSE POSITIVE RATE (FPR) FOR A DAY WITHOUT ATTACKS WHEN SENSORS ARE ACTIVATED ONE AT A TIME. SAMPLING RATE IS FIXED AT 1 MINUTE.

	Uninformed		Informed		Normal
	DT	*DR*	*DT*	*DR*	*FPR*
app	183 s	100.00%	1290 s	80.72%	40.98%
wifi	183 s	100.00%	1825 s	9.03%	28.10%
noise	1020 s	59.64%	Undetected		0.66%
acc	Undetected		Undetected		0.58%
mag	Undetected		Undetected		1.83%
rot	593 s	94.73%	Undetected		5.88%
light	Undetected		3686 s	6.02%	40.98%
amb	183 s	97.36%	Undetected		1.10%
beh	6233 s	13.15%	1825 s	3.61%	1.03%
all	183 s	92.07%	1657 s	28.82%	1.39%

Results indicate that no one sensor provides a good detector on its own. While app and wifi sensors provide reasonable

detection rates under uninformed attack scenarios, they also have high false positive rates under normal use. Similarly, accelerometer, rotation and magnetic field produce low false positive rates, though they are not particularly effective in detecting the attacks. Using ambient and behavioural sensor subsets improve both detection and false positive rates and using the entire set of sensors produce the best results by tempering the extreme positive and negative feedback that might come from fewer sensors. Though while data from multiple sensors might maintain low false positive and high detection rates, it increases the battery consumption of the device.

Thus, there there is no "one size fits all" solution, as using the same sensor contributions and rates of sampling does not take factors such as a user's behaviour and the nature of an attack into account. However, an adaptive solution could control the trade-off between the battery consumption and the detection capabilities. We explore the effectiveness of such a solution in Section V.

V. Adaptive Sampling

The results of Sections III and IV establish the trade-off between battery consumption and detection efficacy. Adaptive sampling can reduce the battery consumption by adjusting the sampling rate and activating high drain sensors only when they are needed, and we hypothesise that this can be done without significantly impacting security. In this section, we focus on defining and analysing options for the use of an adaptive sampling rate.

We define *expected vigilance level* to be a state which sets the sampling rate based upon conditions that could impact device security, and is determined by the device. Needless to say, establishing such a vigilance level requires care as it might be exploited to launch attacks if the sampling rate can be reduced by the attacker. We propose several techniques to determine the vigilance level and adjust the sampling rate of the device that are based on a number of factors including detection score, location and time of day (others are possible). Thus, we examine the following adaptive techniques to determine the vigilance level; the first two are based upon the device-computed detection score, while the latter two are based upon external variables, location and time-of-day.

1) **Change in detection score:** This technique increases the sampling rate of the device, if the difference in detection score (a value between -1 and +1 in *user-profiler*) increases by more than a threshold between current and previous state. The difference in threshold is empirically determined for the test user, but in general, this technique aims to reduce the sampling rate if the detector perceives the conditions to be stable and normal. Needless to say, if the detection score indicates an anomaly, the highest sampling rate is utilised. This technique presents vulnerabilities if the attacker can influence the sensors to stabilise for example, by influencing the sensor data.
2) **Detection score level:** This technique determines the sampling rate based on detection scores. Depending on the detection score, sampling rates are adjusted, e.g., high score (suggesting normal use) reduces the sampling rate. As opposed to the previous technique, this technique determines the sampling rate from the current observation only without taking the trend into account. Similar to the previous technique, it is vulnerable if the attacker can influence sensor data.
3) **Context based:** This technique determines the sampling rate based on the user's whereabouts. Different sampling rates are selected for frequently visited locations (home and work in our experiments). For any other location the sampling rate is selected to be the highest. As opposed to the previous techniques, this does not take detection score into account but makes decisions based on location only. Thus, it is vulnerable to attacks at known locations for which the sampling rate is low. For our study, three leniency levels related to context are defined (see Table VI).
4) **Hourly:** This technique determines the sampling rate based on the hour of day. Higher sampling rates are selected for active hours. However, this technique is vulnerable to attacks at inactive hours. For our study, three leniency levels related to time-of-day are defined (see Table VI).

Table VI summarises the adaptive techniques that determine the vigilance level. In our experiments, the device can choose from the following sampling rates: 1, 2, 5 and 10 minutes. This set provides balanced options from both high and low sampling rates although providing more options may help further optimise battery consumption and detection results. We also compare the adaptive techniques with a baseline fixed sampling rate of 1 minute.

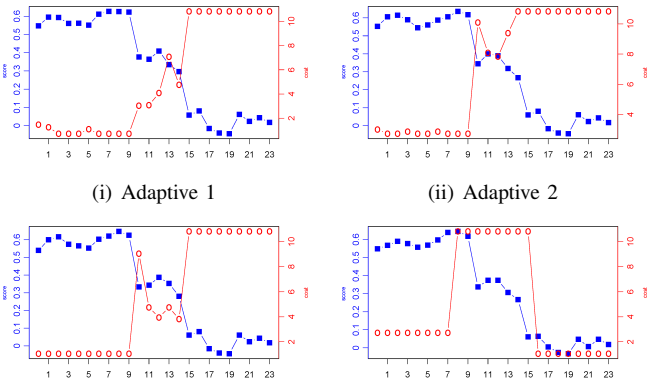
TABLE VI. Summary of adaptive techniques that determine the vigilance level. Options (a) to (c) provide leniency levels, in which (a) is the most strict.

	Technique and description
0	fixed sampling rate=1 at all times (baseline)
1	if $\text{score}(t_n) - \text{score}(t_{n-1}) > 0.5$: increase rate if $0.1 < \text{score}(t_n) - \text{score}(t_{n-1}) < 0.5$: maintain rate else: decrease rate where rate $\in \{1, 2, 5, 10\}$ and $\text{rate}(t_0) = 1$
2	if $\text{score}(t_n) > 0.9$: rate=10 if $\text{score}(t_n) > 0.5$: rate=5 else: rate=1
3a	if at home: rate=5, if at work: rate=2, else: rate=1
3b	if at home: rate=10, if at work: rate=5, else: rate=1
3c	if at home: rate=10, if at work: rate=5, else: rate=2
4a	if morning: rate=2, if noon: rate=1, night: rate=5
4b	if morning: rate=5, if noon: rate=1, night: rate=10
4c	if morning: rate=5, if noon: rate=2, night: rate=10

Table VII (as well as Figures 2, 3, and 4) details the detection and battery consumption results for different adaptive techniques as well as the fixed sampling rate of 1 minute. The results show that the battery consumption can be improved substantially whilst maintaining the detection time. For example, Adaptive 1 provides detection rates as good as the baseline while reducing the consumption from 10.83 mAh to 1.54 mAh on a normal day and to 5.34 mAh under the uninformed attack scenario. On days in which the detection score is low (e.g., uninformed attack which occurred from 2 pm onwards) the battery consumption increases. This is especially evident in Figure 2(i) where the sampling rate increases as the detection score drops.

Figures 2, 3, and 4 show the changes in detection scores (blue lines with squares) and battery cost (red lines with circles) for four of the adaptive techniques from Table VI, respectively for the *uninformed attack*, *informed attack*, and *normal use cases*.

For Figure 2, Adaptive 1 and 2 (Figures 2(i) and 2(ii)) techniques sharply increase the sampling rate when the uninformed attack starts at 2 pm (the uninformed attack continues indefinitely), reflecting a quick detection of this attack based upon changes in the detection scores or score level, respectively. Adaptive 3b and 4b (Figures 2(iii) and 2(iv)) techniques use location and time to determine the sampling rate, and show similar performance with a lower detection score, and higher battery consumption when under attack. However, note that for the Adaptive 4b technique, there is a vulnerability in this case as the sampling rate is lowered in the evening, even though the device was still in the attacker's control.



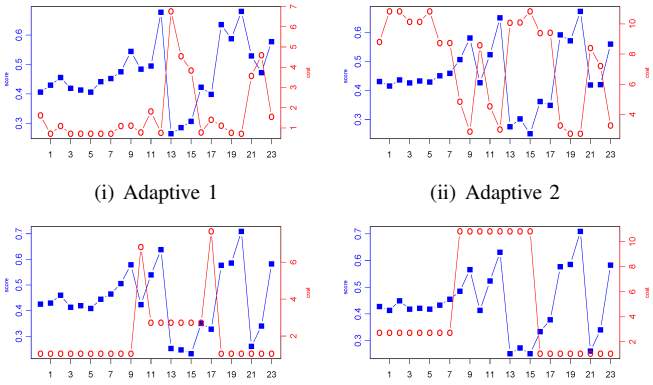
(i) Adaptive 1 (ii) Adaptive 2

(iii) Adaptive 3b (iv) Adaptive 4b

Fig. 2. Detection scores and average battery consumption for uninformed attack. X axis shows hours, red lines with circles show mean battery consumption for the hour and blue lines with squares show the mean detection score for the hour.

Figure 3 shows the detection scores and battery consumption for the informed attack scenario (red lines with circles = battery consumption; blue lines with squares = detection score). Adaptive 1 and 2 (Figures 3(i) and 3(ii)) increase the sampling rate (hence the cost) between 1 pm and 5 pm when attack is taking place, and return to normal after the temporary attack. Adaptive 2, maintains a high sampling rate even before the attack because the detection scores for the day were not high enough to reduce the sampling rate. Thus, Adaptive 2 can potentially increase the battery consumption on days when many unusual events (but legitimate deviations from routine) are observed. Unlike the case of the vulnerability for the unformed attack discussed in Figure 2(iv) above, for the informed attack the Adaptive 4b technique provides good detection results since the sampling rate coincidentally happened to be higher prior to the attack.

In contrast, Figure 4 shows the detection scores and battery consumption for the normal day (red lines with circles = battery consumption; blue lines with squares = detection score). Adaptive 1 and 2 techniques (Figures 4(i) and 4(ii)) adjust the sampling rate based on user's behaviour, increasing the sampling rate as the detection score drops. Their patterns differ as Adaptive 1 looks for the detection score to stabilise whereas Adaptive 2 looks for the score to be high. Thus, on this

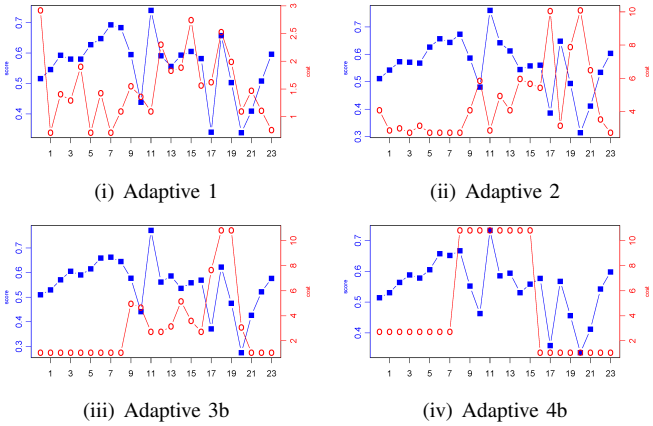


(i) Adaptive 1 (ii) Adaptive 2

(iii) Adaptive 3b (iv) Adaptive 4b

Fig. 3. Detection scores and average battery consumption for informed attack. X axis shows hours, red lines with circles show mean battery consumption for the hour and blue lines with squares show the mean detection score for the hour.

normal day where detection score is stable, Adaptive 1 reduces the battery consumption from 10.83 mAh to 1.54 mAh (see Table VII). Such a reduction would have a substantial impact on light and medium drain users. Referring to Tables I and II, a consumption reduction from roughly 10 mAh to 1 mAh can double the battery life for light drain users. Adaptive 3 and 4 (Figures 4(iii) and 4(iv)) are once again respectively dependent upon changes in location and time of day. Interestingly in this case, the sampling rate (and hence battery consumption) for Adaptive 4 remains low even though there are some clear drops in the detection scores at around 5pm and 8pm.



(i) Adaptive 1 (ii) Adaptive 2

(iii) Adaptive 3b (iv) Adaptive 4b

Fig. 4. Detection scores and average battery consumption for normal use. X axis shows hours, red lines with circles show mean battery consumption for the hour and blue lines with squares show the mean detection score for the hour.

From these example attack scenarios and candidate adaptive techniques, we can highlight some general observations. Referring to Table VII, Adaptive 3a, 3b and 3c detected the attacks later because the attacks either started (in case of uninformed scenario) or occurred (in the informed scenario) at a location which the scheme deemed as 'safe'. This shows a potential vulnerability in context based adaptation, in which the attacker may easily coerce the device to sample less by taking it to a location familiar to the device. Similarly, Adaptive 4a, 4b and 4c suffers from a similar vulnerability in which the attacker may choose to attack the device when they know the sampling

rate is lower. Adaptive 1 and 2 on the other hand suffers from a possible 'replay attack' where the attacker can try to alter the ambient and some behavioural characteristics to influence the detection score, causing the device to sample less before carrying out their attack. This provides some more empirical evidence confirming that there is no "one solution for all" in this case, though smart selection of the potential techniques, dependent upon factors such as the user's behaviour, and location and time context, would seem to be a viable path for continued exploration.

VI. Conclusion

In this paper, we conducted a cost/benefit analysis of sensors that are commonly found on today's mobile devices. To this end, we computed the sampling costs of sensors and established their usefulness in a data-driven authentication scenario. While previous work investigated the sensor costs, we are – to the best of our knowledge – the first to investigate this phenomenon in the context of data-driven authentication. Given a large array of sensors that user behaviour modelling techniques can use, we believe it is important to identify the sensors that provide best support for detecting attacks and to establish their sampling costs.

Our battery consumption results in Section III indicate that with high sampling rates, light and medium drain users are most impacted by sensor costs, but that rate reductions can significantly reduce consumption. However, detection results under different sampling rates in Section IV showed that sampling rates over a few minutes would not be effective against attacks. When we analysed the sensor data individually we identified that using many sensors at once tempers the extreme values from a single sensor and reduces false positives. In the light of the battery consumption results and per sensor analysis, we proposed various adaptive sampling techniques in Section V that determine the vigilance level based on detection score, location and time. Adaptive sampling results indicate that it is possible to reduce the battery consumption from about 10 mAh to 1 mAh without impacting the detection of attacks.

Needless to say, measuring battery consumption and usefulness is not without its pitfalls. Differences in hardware, software and the physical environment are some of the factors that can affect the battery consumption. Our results provide consumption and detection results for the *collector* and *user-profiler*. While we believe it provides a good estimate of each sensor's battery drain and its usefulness, using different collection and detection techniques is likely to affect the detection and consumption results. Our field data is limited to a relatively small number of users and the attack data is collected from a single device. However, we highlight that our objective is not to establish exact battery consumption and usefulness metrics but rather to encourage adaptivity in design where sampling rate does not have to be fixed and not all sensors cost and contribute equally.

Our future work will focus on extending our study to a wider range of sensors and with more users. We also aim to investigate an adaptive modelling technique which utilises the sensors selectively based on device context hence using high benefit, low cost sensors first. This will allow a device to leverage battery consumption costs and detection results to learn the optimum strategy in terms of which sensors to use and when and how frequently to sample.

Acknowledgment

The research leading to these results has received funding from the People Programme (Marie Curie Actions) of the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no PIIF-GA-2011-301536. We also thank Mobolaji Ayoade for his help with battery consumption experiments.

References

[1] D. Ferreira, E. Ferreira, J. Goncalves, V. Kostakos, and A. K. Dey, "Revisiting human-battery interaction with an interactive battery interface," in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '13. New York, NY, USA: ACM, 2013, pp. 563–572.

[2] Cell phone battery statistics across major US cities. [Online]. Available: http://www.prweb.com/releases/2013/12/prweb11387942.htm

[3] A. Gupta, M. Miettinen, N. Asokan, and M. Nagy, "Intuitive security policy configuration in mobile devices using context profiling," in *Proceedings of the 2012 ASE/IEEE International Conference on Social Computing*. IEEE CS, 2012, pp. 471–480.

[4] E. Shi, Y. Niu, M. Jakobsson, and R. Chow, "Implicit authentication through learning user behavior," in *Proceedings of the 13th international conference on Information security*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 99–113.

[5] M. Tamviruzzaman, S. I. Ahamed, C. S. Hasan, and C. O'Brien, "epet: when cellular phone learns to recognize its owner," in *Proceedings of the 2nd workshop on Assurable and usable security configuration*. ACM, 2009, pp. 13–18.

[6] O. Riva, C. Qin, K. Strauss, and D. Lymberopoulos, "Progressive authentication: Deciding when to authenticate on mobile phones," in *Proceedings of the 21st USENIX Conference on Security Symposium*. USENIX Association, 2012, pp. 15–15.

[7] H. G. Kayacik, M. Just, L. Baillie, D. Aspinall, and N. Micallef, "Data driven authentication: On the effectiveness of user behaviour modelling with mobile device sensors," in *Proceedings of the Mobile Security Technologies Workshop*, 2014.

[8] K. Lin, A. Kansal, D. Lymberopoulos, and F. Zhao, "Energy-accuracy trade-off for continuous mobile device location," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. ACM, 2010, pp. 285–298.

[9] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The jigsaw continuous sensing engine for mobile phone applications," in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2010, pp. 71–84.

[10] P. Wu, J. Zhu, and J. Y. Zhang, "Mobisens: A versatile mobile sensing platform for real-world applications," *Mobile Networks and Applications*, vol. 18, no. 1, pp. 60–80, 2013.

[11] J. Paek, J. Kim, and R. Govindan, "Energy-efficient rate-adaptive GPS-based positioning for smartphones," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. ACM, 2010, pp. 299–314.

[12] Z. Zhuang, K.-H. Kim, and J. P. Singh, "Improving energy efficiency of location sensing on smartphones," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*. ACM, 2010, pp. 315–330.

[13] Y. Wang, J. Lin, M. Annavaram, Q. A. Jacobson, J. Hong, B. Krishnamachari, and N. Sadeh, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*. ACM, 2009, pp. 179–192.

[14] M. Schirmer and H. Höpfner, "SENST*: Approaches for reducing the energy consumption of smartphone-based context recognition," in *Proceedings of the 7th International and Interdisciplinary Conference on Modeling and Using Context*. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 250–263.

TABLE VII. DETECTION TIME (DT) AND RATE (DR) OF ATTACKS, FALSE POSITIVE RATE (FPR) FOR A DAY WITHOUT ATTACKS AND BATTERY CONSUMPTION (BC) FOR DIFFERENT ADAPTIVE SAMPLING TECHNIQUES.

Technique	**Uninformed**			**Informed**			**Normal**	
	DT	*DR*	*BC*	*DT*	*DR*	*BC*	*FPR*	*BC*
0	183 s	92.07%	10.83 mAh	1657 s	28.82%	10.83 mAh	1.39%	10.83 mAh
1	183 s	97.37%	5.34 mAh	1206 s	36.48%	1.75 mAh	3.15%	1.54 mAh
2	183 s	92.07%	7.14 mAh	1507 s	30.39%	7.50 mAh	3.11%	4.55 mAh
3a	774 s	95.04%	6.75 mAh	1963 s	29.72%	3.84 mAh	2.32%	4.78 mAh
3b	3484 s	96.78%	5.59 mAh	3489 s	31.11%	1.97 mAh	1.63%	3.10 mAh
3c	3709 s	95.53%	3.19 mAh	3489 s	31.11%	1.69 mAh	1.92%	2.21 mAh
4a	183 s	78.43%	6.32 mAh	1657 s	30.33%	6.32 mAh	1.04%	6.32 mAh
4b	183 s	72.15%	4.86 mAh	1657 s	30.23%	4.86 mAh	0.68%	4.86 mAh
4c	774 s	79.41%	3.05 mAh	1963 s	31.46%	3.05 mAh	1.66%	3.05 mAh

[15] X. Li, H. Cao, E. Chen, and J. Tian, "Learning to infer the status of heavy-duty sensors for energy-efficient context-sensing," *ACM Trans. Intell. Syst. Technol.*, vol. 3, no. 2, pp. 35:1–35:23, Feb. 2012.

[16] H. Hopfner and K.-U. Sattler, "Cache-supported processing of queries in mobile DBS." in *Database Mechanisms for Mobile Applications*, vol. 43, 2003, pp. 106–121.

[17] S. Nath, "Ace: Exploiting correlation for energy-efficient and continuous context sensing," in *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*. ACM, 2012, pp. 29–42.

[18] A. Krause, M. Ihmig, E. Rankin, D. Leong, S. Gupta, D. Siewiorek, A. Smailagic, M. Deisher, and U. Sengupta, "Trading off prediction accuracy and power consumption for context-aware wearable computing," in *Proceedings of the IEEE International Symposium on Wearable Computers*. IEEE Computer Society, 2005, pp. 20–26.

[19] K. Rachuri, C. Mascolo, and M. Musolesi, "Energy-accuracy trade-offs of sensor sampling in smart phone based sensing systems," in *Mobile Context Awareness*, T. Lovett and E. O'Neill, Eds. Springer London, 2012, pp. 65–76.

[20] C.-C. Lin, D. Liang, C.-C. Chang, and C.-H. Yang, "A new non-intrusive authentication method based on the orientation sensor for smartphone users," in *Proceedings of the IEEE International Conference on Software Security and Reliability*, 2012, pp. 245 –252.

[21] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong, "Senguard: Passive user identification on smartphones using multiple sensors," in *Proceedings of the 7th International Conference on Wireless and Mobile Computing, Networking and Communications*, 2011, pp. 141 –148.

[22] E. Hayashi, S. Das, S. Amini, J. Hong, and I. Oakley, "Casa: Context-aware scalable authentication," in *Proceedings of the Ninth Symposium on Usable Privacy and Security*. ACM, 2013, pp. 3:1–3:10.

[23] I. T. Fischer, C. Kuo, L. Huang, and M. Frank, "Smartphones: Not smart enough?" in *Proceedings of the Workshop on Security and Privacy in Smartphones and Mobile Devices*. ACM, 2012, pp. 27–32.

[24] J. Seifert, A. De Luca, B. Conradi, and H. Hussmann, "Treasurephone: Context-sensitive user data protection on mobile phones," in *Proceedings of the 8th International Conference on Pervasive Computing*. Springer-Verlag, 2010, pp. 130–137.

[25] L. Zhang, B. Tiwana, Z. Qian, Z. Wang, R. P. Dick, Z. M. Mao, and L. Yang, "Accurate online power estimation and automatic battery behavior based power model generation for smartphones," in *Proceedings of the 8th IEEE International Conference on Hardware/Software Codesign and System Synthesis*. ACM, 2010, pp. 105–114.

[26] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin, "Diversity in smartphone usage," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '10. New York, NY, USA: ACM, 2010, pp. 179–194.

[27] Userprofiler project. [Online]. Available: https://github.com/kayacik/userprofiler